UNITED STATES PATENT APPLICATION

for

A MECHANISM TO REPEAT SIGNALS ACROSS AN UNRELATED LINK

Inventors:

Dean T. Lindsay

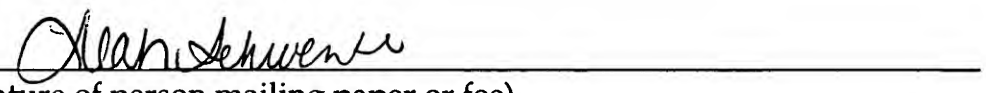
Prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026
(303) 740-1980

File No.: 042390.P18576

"Express Mail" mailing label number EV306655089US
Date of Deposit March 29, 2004
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, P.O. Box 1450, Alexandria, VA 22313-1450.

Leah Schwenke
(Typed or printed name of person mailing paper or fee)


(Signature of person mailing paper or fee)

A MECHANISM TO REPEAT SIGNALS ACROSS AN UNRELATED LINK

FIELD OF THE INVENTION

[0001] The present invention relates to integrated circuits; more particularly, the present invention relates to interfacing integrated circuits.

BACKGROUND

[0002] Integrated circuits (ICs) are commonly connected together into a system by interfaces for data transfer and control. Such interfaces may include buses and point-to-point links. Typically, there are additional signals, not directly related to the links, which are implemented to coordinate the actions of the ICs. Such signals include error and debug signals, among other types of signals.

[0003] When these signals are run as physical wires, each separate wire must have its own pins, timing specs, voltage specs, etc., which complicates the IC transmitting and the IC receiving the signal. Current systems implement unassigned protocol points on the existing buses or links connecting the ICs to communicate the same information between ICs which would have been carried on discrete wires, thus simplifying the physical interfaces between ICs by eliminating the additional wires.

[0004] Such a mechanism is referred to as in-band signaling because the additional information, although not protocol related, is carried as part of the

protocol signaling, and thus is in-band to the protocol. For example, one protocol point might indicate that "debug signal A asserted." Another protocol point would indicate that "debug signal A deasserted". When a first IC needs to communicate that debug signal A has been asserted to a second IC, the message "debug signal A asserted" is transmitted in the ongoing protocol stream from the first IC to the second IC. When debug signal A deasserts, the first IC sends "debug signal A deasserted" to the second IC.

[0005] Although this mechanism of conveying signals between ICs without adding additional wires provides many benefits, there are also significant disadvantages, such as when a number of signals are being transmitted between ICs in this manner.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention. The drawings, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

[0007] **Figure 1** illustrates one embodiment of a computer system;

[0008] **Figure 2** illustrates one embodiment of a transmitting IC;

[0009] **Figure 3** illustrates one embodiment of a receiving IC;

[0010] **Figure 4** illustrates another embodiment of a transmitting IC;

[0011] **Figure 5** is a timing diagram showing one embodiment of signal timing; and

[0012] **Figure 6** is a timing diagram showing another embodiment of signal timing.

DETAILED DESCRIPTION

[0013] A mechanism to repeat signals across an unrelated link is described. In the following description, numerous details are set forth. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

[0014] Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment.

[0015] **Figure 1** is a block diagram of one embodiment of a computer system 100. Computer system 100 includes a central processing unit (CPU) 102 coupled to bus 105. In one embodiment, CPU 102 is a processor in the Pentium® family of processors including the Pentium® II processor family, Pentium® III processors, and Pentium® IV processors available from Intel Corporation of Santa Clara, California. Alternatively, other CPUs may be used.

[0016] A chipset 107 is also coupled to bus 105. Chipset 107 includes a memory control hub (MCH) 110. In one embodiment, MCH 110 is coupled to an input/output control hub (ICH) 140 via a hub interface. ICH 140 provides an

interface to input/output (I/O) devices within computer system 100. For instance, ICH 140 may be coupled to a Peripheral Component Interconnect bus adhering to a Specification Revision 2.1 bus developed by the PCI Special Interest Group of Portland, Oregon.

[0017] In one embodiment, MCH 110 includes a memory controller 112 that is coupled to a main system memory 115. Main system memory 115 stores data and sequences of instructions and code represented by data signals that may be executed by CPU 102 or any other device included in system 100. In one embodiment, main system memory 115 includes dynamic random access memory (DRAM); however, main system memory 115 may be implemented using other memory types.

[0018] According to one embodiment, MCH 110 and ICH 140 are implemented on separate ICs that are coupled via the hub interface. However, there may be additional signals (e.g., error signals, debug signals, etc.) not related to the hub interface that may be needed to coordinate MCH 110 and ICH 140. These signals generally have the following characteristics: 1) transitions can occur on any clock cycle; 2) limited latency is permissible between the time when a signal asserts and when the signal is observed by another IC; 3) the timing of the first transition from steady state is important, signaling the beginning of a condition, whereas the timing of the return-to-steady-state transition is less important; 4) each signal changes infrequently, but multiple signals may change on nearby clock cycles, and preserving the relative timing of the signal changes

to each other is important.

[0019] As discussed above, in-band signaling is an existing mechanism implemented to communicate information between ICs. A disadvantage occurs using in-band signaling when multiple signals passed using in-band signaling change state at the same or nearly the same time (related to the speed of signal transmission by the protocol). In such a case, the protocol serializes the signal information, transmitting the new state of one signal, and then later transmitting the new state of the next signal, until the changes on all signals are communicated.

[0020] The result is that actions seen at essentially the same time in one IC are seen sequentially in time at the other IC. For example, suppose that a first IC (IC 1) and a second IC (IC 2) are connected by a link with a communicating protocol. Suppose that the values of two signals, A and B, are of interest to both ICs. The signals exist in IC 1 and will be communicated in-band across the link to IC 2, although the signals are not related to the operation of the link. When A alone asserts in IC 1, this is signaled to IC 2 with the "A asserted" message, and IC 2 knows that A asserted within 1 link latency of when it occurred. The same occurs when signal B alone asserts, and IC 2 knows that B asserted within 1 link latency of when it occurred. However, when A and B assert simultaneously, this cannot be communicated simultaneously, as "A asserted" and "B asserted" are distinct protocol points that cannot be simultaneously communicated.

[0021] Consequently, some mechanism must prioritize one signal over the

other and serially transmit the state changes. The state change incurred by the signal that is not transmitted must somehow be stored and transmitted at the next opportunity. Thus, the information known in IC 1 that A and B asserted simultaneously is temporally distorted by arriving in IC 2 as A asserted, followed a link latency later by B asserted. This temporal distortion in IC2 is illustrated in **Figure 5**.

[0022] The above described example would be further complicated if signals A and B assert and deassert simultaneously, and changes on signal A have priority over changes on signal B. This situation is illustrated in **Figure 6**. "A asserted" is chosen to be transmitted first. On the next clock cycle of IC 1, while "A asserted" is being sent, A and B deassert. Now "A deasserted" is chosen to be transmitted. "B asserted" is thus delayed two link message times.

[0023] In addition, there are two entries required in the buffer that is remembering to transmit "B asserted" followed by "B deasserted." The buffer can become full and overflow, and the associated error conditions must be handled. The exception conditions to handle the buffers are complicated and prone to design errors. It is therefore possible for the final signal state after a series of transitions to be lost in the buffers so that the receiving IC at the end has the wrong state of one or more of the signals.

[0024] Currently, two points are assigned in the protocol per signal, one for assertion and another for deassertion. Thus, 2*N protocol points for N in-band signals are currently implemented. According to one embodiment,

protocol points for in-band signal transmissions are changed. In such an embodiment, the state of all in-band signals are transmitted each time that any change is observed on any of the in-band signals. Consequently, 2^n protocol points are implemented for N in-band signals (e.g., one protocol point for each possible condition of the N signal lines being transmitted in-band).

[0025] In a further embodiment, the state of all in-band signals being transmitted is sampled whenever any of the signals change. Subsequently, the state of all of the signals is transmitted at once in a single protocol point across the interface. In yet a further embodiment, the IC receiving the signals drives all signals with the new values it receives each time an in-band protocol point is received across the interface.

[0026] **Figure 2** illustrates one embodiment of a transmitting IC 200. In one embodiment, transmitting IC 200 is MCH 110. However, one of ordinary skill in the art will appreciate that transmitting IC 200 may be any other type of IC coupled to another IC via any type of bus or interface. Referring to **Figure 2**, IC 200 includes signal logic 210 and protocol logic 230.

[0027] In one embodiment, IC 200 includes signal logic 210 for each signal that is to be transmitted to a receiving IC. Thus, signal logic 210(1) - 210(n) is included corresponding to signals A - N, where N represents the Nth signal and n represents the corresponding signal logic 210. Signal logic 210 includes the logic that enables each separate signal to be repeated in-band across an interface. Protocol logic 230 is the logic that receives the signals to be repeated in-band and

selects an appropriate protocol point and integrates the protocol point into a protocol that is being transmitted across the interface.

[0028] According to one embodiment, all signals shown and flip-flops within signal logic 210 and protocol logic 230 operate on a common clock. The description below will focus on signal A and corresponding signal logic 210, with other signals operating in a similar fashion. After initialization, the flip-flop 2 (FF2) in signal logic 210, which drives a send signal, is cleared. This allows signal A to propagate through the multiplexer (mux) to the D input of FF1. One clock cycle later a signal_A_held signal has the same value as signal A. This is the steady-state condition of the circuit.

[0029] A change from steady state occurs when signal A asserts after being unasserted for a long time. When signal A asserts, the XOR gate sees different values on its inputs, and its output (L) is asserted. This asserts the D input of a FF2 through the OR gate, which generates a send signal. On the next clock, both the send signal and signal_A_held are asserted. Since both signal A and signal_A_held now have the same value, the output of the XOR gate L is now deasserted.

[0030] At the same time, the send signal has become high and because the sent signal was already steady-state low, the AND gate asserts its output to the OR gate, which maintains its output, asserting the D input to FF2. A feedback loop is thus formed from the Q to the D of FF2, which maintains its output constantly asserted as long as the sent signal is low.

[0031] Assertion of the send signal also switches the mux so that a feedback loop is formed from the Q output to the D input of FF1. As long as the send signal is asserted, signal A is ignored and the signal_A_held signal maintains the value that it had when the send signal was asserted. As will be discussed below, the send signal remains in this state until the new value of signal A has been repeated across the interface. Thus the mux serves the purpose of ignoring those signal transitions, that occur to close to a previous transition to be repeated across the interface. This is accomplished without the need to buffer signal changes and then discard buffer contents when interface bandwidth does not allow them to be transmitted, as is the case with the prior art.

[0032] For protocol logic 230, the values of all signal_X_held signals are observed continuously. Protocol logic 230 also observes the OR of the send signals from each signal logic 210 component. When any send signal asserts, the send input at protocol logic 230 asserts. When the send signal from signal logic 210 corresponding to signal A asserts the send signal at protocol logic 230, protocol logic 230, at the next available opportunity, places the values of all of the signal_X_held signals into a single in-band signaling protocol frame using the protocol point that represents all of their immediate values. Then the sent signal is asserted for a single clock cycle.

[0033] When the sent signal arrives at signal logic 210 for signal A, the feedback loop from Q to D for FF2 is broken, and the D input is unasserted. At

the next clock cycle, the send signal is deasserted, which switches the mux so that signal A now propagates to the D input of FF1 and to the input to the XOR gate.

[0034] Assuming that signal A is still in the same state that was just repeated (or has returned to that state again), the XOR sees the same values on its two inputs, and the circuit has returned to steady-state. However, should signal A now have a different value, the XOR gate detects this and the process described above repeats to transmit this new value across the interface.

[0035] It can be seen that when N signals are being repeated in-band across the interface, that they may be changing at any clock cycle relative to each other. On some clock cycles there will be no changes, but there may be one or several changes on other clock cycles. Also, the latency until the repeating interface is able to repeat the changes may be short or long, constant or varying, depending upon the interface design. In any case, the first change causes the interface to prepare to repeat signals.

[0036] During the time required by the interface to prepare to repeat the signals, other signals may change state, and their new state will also be repeated once preparation is complete. When the interface is ready, it transmits the states of all signals, including the new state of each signal that has changed up to that time, across the interface at once. The signals are then reevaluated for changes relative to the value that was just repeated across the interface. When new changes are detected the process is repeated.

[0037] **Figure 3** illustrates one embodiment of a receiving IC 300. IC 300

includes protocol logic 330 and sequential logic 350. Sequential logic 350 includes a FF dedicated for each repeated signal that is being received. When protocol logic 330 detects an incoming in-band signaling protocol frame, protocol logic 330 extracts the state of all of the received signals and presents this to the D inputs of the FFs at sequential logic 350, at the same time enabling the FFs to accept data with an enable signal. These FFs maintain the repeated signals at the last-written values until the next in-band signaling protocol frame causes them to be updated.

[0038] Increasingly, IC links are high speed, and are subject to transmission errors. In-band signal repeating is sometimes inserted into the lower protocol layers for reasons such as minimization of transmission latency and reduction of buffer complexity at higher protocol levels. The result is that in-band signal repeating frames may be lost.

[0039] The above-described repeating mechanism transmits one signal-repeating frame for all signal changes that fall within the frame's time window. If that frame is lost, the receiving IC has the wrong state for all signals with changes contained within the lost frame. This condition will persist until another signal changes, and all signal states are again repeated.

[0040] A degree of tolerance to loss of in-band signaling protocol frames can be achieved by a simple change to signal logic 210 described above. The change causes multiple frames to be transmitted for each signal change, thus increasing the probability of delivery of the signal change to the receiver.

[0041] The number of frames to be transmitted for each signal change can be set to accommodate the expected error rate on the physical link. Transmitting multiple frames for each change uses link bandwidth, but does not require buffering or retry schemes to be implemented. Latency to repeat the signals across the link is minimized because the first correct frame to arrive sets the signal values at the receiver.

[0042] **Figure 4** illustrates another embodiment of a transmitting IC 200 including an in-band signaling mechanism that is tolerant of the loss of in-band signaling protocol frames. To implement transmitting multiple frames for each signal change in signal logic 210, the logic within the dotted block (e.g., the Or-gate, And-gate and FF2) is replaced by a loadable countdown counter 420. The signal L is the load signal for counter 420. When L is asserted, counter 420 loads its default value. When signal E is asserted, counter 420 is enabled to count down. For every clock cycle that C is asserted when E is also asserted, counter 420 decrements by one count. Counter 420 output N is asserted whenever the counter value is non-zero.

[0043] Counter 420 operates in IC 200 as follows. When the XOR gate asserts as a result of a signal change, L is asserted. L is asserted for only one cycle, as the first clock cycle after the signal change makes the XOR inputs the same. Asserting L loads the default value into the counter on the next clock edge. The default counter 420 value is chosen to balance the link error rate, the impact on the protocol throughput, the importance of the repeated data, etc.

[0044] When counter 420 is loaded, its N output asserts because counter 420 is now non-zero. This switches the mux, holding the value of the input signal while it is transmitted. Asserting the N output also asserts signal E, enabling counter 420 to count down based on the assertion of the C input. It also asserts the send signal to protocol logic 230.

[0045] Each time that the protocol logic 230 observes the send signal asserted, protocol logic 230 prepares to repeat the signals, captures their values, sends an in-band signaling protocol frame, and asserts the sent signal. The counter 420 in each signal logic 210 component that has experienced a signal change has a non-zero value. The sent pulse decrements each counter 420 in this condition by one count, indicating that the signal change this counter is tracking has been transmitted once over the link. This continues until each such counter reaches 0, *N* deasserts, counter 420 is disabled (so that it does not underflow), the send signal from signal logic 210 to the protocol logic is deasserted, and the mux switches to watch for further input signal changes.

[0046] The above-described mechanism transmits the default number of in-band signal frames for each signal transition. When changes on additional signals occur during the repeating of earlier changes on other signals, these additional changes are immediately incorporated into the next and subsequent in-band signaling frames being sent. The transmission of in-band signaling protocol frames stops once the last-occurring signal change has been repeated the default number of times

[0047] The receiving IC 300 (**Figure 3**) used with the signal logic 210 including a counter is identical to the receiver already described above. In-band signaling protocol frames that are damaged by errors are detected by link checking by protocol logic 330 and are discarded without the enable signal being asserted. Therefore the outputs of the FFs within sequential logic 350, which hold the signal values in IC 300, do not change. The first in-band signaling protocol frame that passes the link checking changes all of the signal FFs to their correct repeated values. Subsequent good frames also write the FFs, but the values are the same so no change is perceived in the signals in IC 300. As long as a single good frame is received, the signals will be correctly repeated.

[0048] The above-described mechanisms enable simultaneous signal transitions in a first IC to be simultaneously observed in a second IC after a transmission delay. Further, the need for complex buffers (error-prone design) to hold transitions for later transmission is obviated. This is true even when errors on the intervening link cause some protocol frames to be lost. Moreover, transitions occurring within a few clock cycles of each other on several signals may be transmitted across an interface without overflowing buffers, and without signal transitions being lost in an unpredictable (or difficult to predict) fashion.

[0049] Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way

intended to be considered limiting. Therefore, references to details of various embodiments are not intended to limit the scope of the claims, which in themselves recite only those features regarded as the invention.